

CORBITS · PRODUCT PACKET · AUGUST 2026

agentflight

The operating system for shipping better Apple apps.

A governed, evidence-first way to build, test, release, and grow Apple-platform products—powered by open-source Interchange.

Prepared for Alex Kehaya, Brian Fox, and the Corbits team
Vision and target architecture. Not a claim of current production availability.

THE CUSTOMER PROBLEM

Apple teams ship across too many disconnected surfaces.

Xcode. GitHub. TestFlight. App Store Connect. Screenshots. Metadata. Analytics. Campaigns. The work is fragmented; the accountability is worse.

01

AI without operational trust

Chat can make suggestions. It rarely proves what changed, who approved it, or whether an external effect actually happened.

02

Automation without context

Scripts can move quickly. They do not naturally understand policy, approvals, evidence, recovery, or a team's actual intent.

03

Release work is still glue work

Teams repeatedly reassemble the same cross-tool operational knowledge for every build, listing, asset, and launch.

04

Trust is the missing layer

The opportunity is not more automation. It is automation a responsible operator can understand, govern, and verify.

THE PROMISE

"Connect an app. In minutes, receive an evidence-backed accessibility assessment. When you choose, AgentFlight prepares a reviewable draft PR."

inspect → evidence → explain → choose scope → test → draft PR

The first product is not "an agent platform." It is a useful, governed improvement against a codebase the customer already cares about.

Start from the outcome, not a terminal or a model picker.

Assess accessibility → Create project → Read-only repo → Evidence report → Verify on Mac

GitHub read-only assessment

Fast source/config/test evidence through a repo-scoped GitHub App. It is visibly labeled static analysis—not runtime VoiceOver verification.

Mac deep assessment

Explicit local enrollment unlocks Xcode, simulator, screenshots, runtime evidence, and local/unpushed project context.

Trust copy must answer four questions

What will happen? Why now? What stays local? How can I revoke it? A permission prompt that cannot answer those questions is rejected.

Homebrew should produce proof, not configuration debt.

```
$ brew install corbits/tap/agentflight
$ agentflight doctor
```

- ✓ AgentFlight CLI 0.1.0 – signed release verified
- ✓ Interchange runtime – open-source dependency available
- ✓ Xcode 26.6 – simulator assessment supported
- Edge not enrolled – no local project access granted

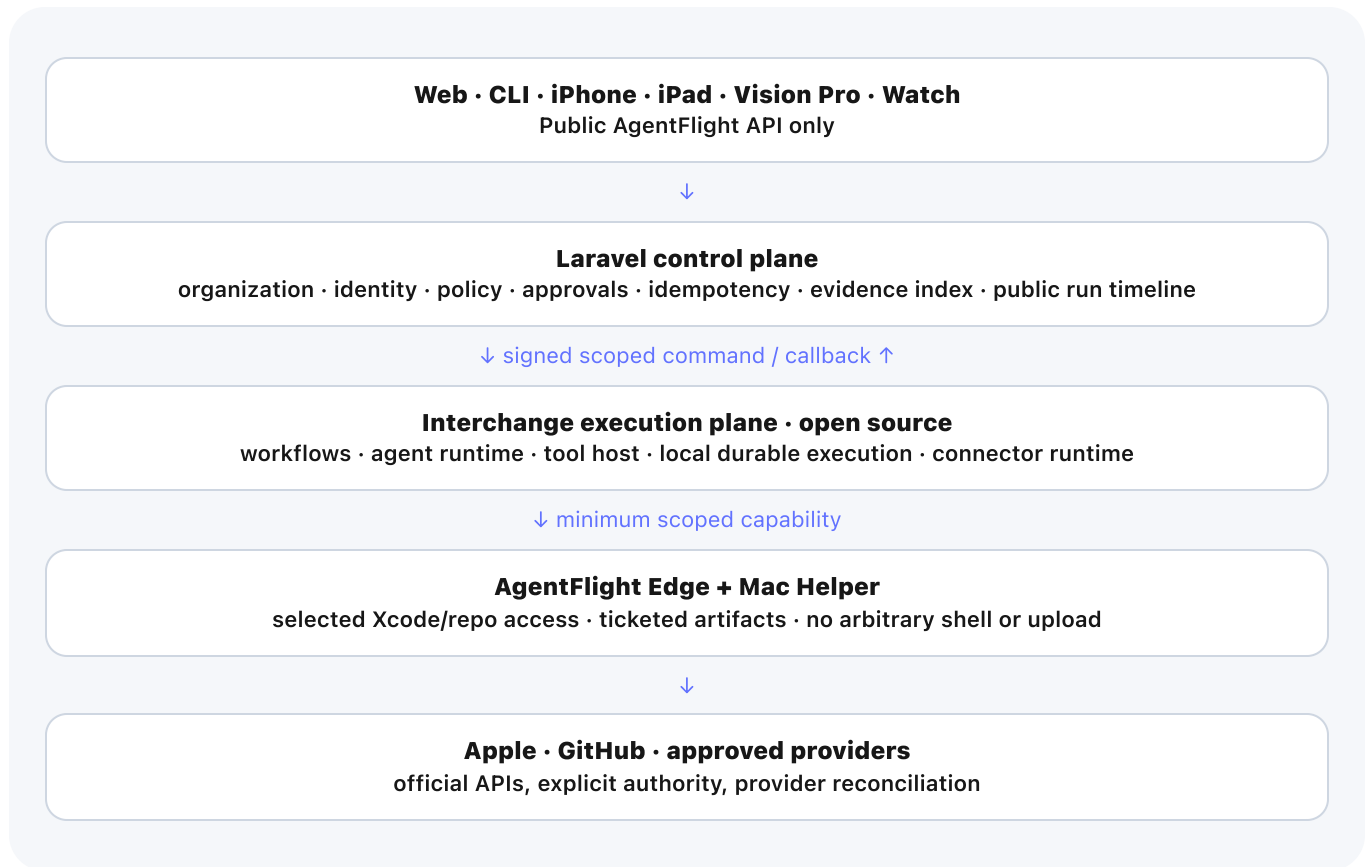
```
$ cd ~/Code/MyApp
$ agentflight assess accessibility --local-preview
```

```
Found 7 reviewable opportunities
3 high confidence · 2 runtime verification recommended
No source uploaded. No branch created. No provider credentials requested.
```

```
$ agentflight login
$ agentflight init
$ agentflight fix plan --finding ax-042
$ agentflight pr create --plan plan-18
```

```
Approval required: draft PR only
Repository: acme/MyApp · Base: main
Diff digest: sha256:... · Tests: 42 passed
[Approve] [Review diff] [Cancel]
```

One product surface. Clear boundaries.



Control plane owns

Identity, policy, approval, evidence, public state, capability issuance, collaboration.

Execution plane owns

Workflow runtime, agent/tool isolation, local execution, connector invocation and signed callbacks.

Interchange stays open. AgentFlight makes it operable at team scale.

Interchange · open source

- agent/workflow runtime
- local tool host
- local durable execution
- open developer interfaces
- customer-authored local workflows

AgentFlight · commercial

- organization/policy/approval system
- managed operation registry and connectors
- evidence, audit, collaboration
- capability issuance and updates
- quality-tested workflow packs

Anything running locally may be inspected. Commercial value lives in the managed service, safety intelligence, collaboration, operational quality, and contractual enterprise offering—not security through obscurity.

More capability, never more unaccountable authority.

Now: Accessibility Steward

Evidence-first accessibility review and bounded draft PR workflow.

Next: Release readiness

Release Captain, Build Inspector, Storefront Curator, Asset Steward, Compliance Sentinel, Evidence Keeper.

Then: Growth intelligence

Market Scout, Channel Analyst, Creative Strategist, Attribution Analyst, Experiment Librarian.

Last: Delivery and spend

Social Publisher, Campaign Operator, Spend Controller—each separately scoped, approved, capped, and reconciled.

Governing loop

prepare → approval → delivery attempt → reconciliation → evidence

Apple quality is not decoration. It is a functional requirement.

● Legibility

One clear next action. State, evidence, and recovery are understandable before any effect occurs.

● Accessibility

VoiceOver, Dynamic Type, contrast, reduced motion, localization/RTL are release gates.

● Privacy

Local/cloud boundaries are explicit. No secret leakage. No surprise source upload.

● Resilience

Offline, reconnect, replay, error, and ambiguous outcomes have useful operator-facing recovery.

The customer should feel that a brilliant Apple-platform operator is beside them—not that a model was unleashed near their production systems.

Move quickly. Make authority unambiguous first.

Gate 0 · Current

Freeze contract truth

Executable plan binding, registry enforcement, lifecycle/recovery state, authority/fencing, generator proof, Edge ownership.

Slice 1

One governed loop

Inspect → plan → approval → narrow TestFlight or PR delivery → reconciliation → evidence.

Slice 2

Release readiness

Build, metadata, assets, compliance, and human-reviewed App Store preparation.

Slice 3+

Intelligence then delivery

Research and planning before public posting, listing changes, and spend controls.

Strategic asks

- Confirm Accessibility Steward → draft PR as the first commercial customer experience.
- Confirm Interchange open core; AgentFlight owns managed policy, evidence, connector intelligence, and collaboration.
- Confirm that contract red-team gates are a prerequisite to broad application implementation.
- Name accountable owners for control plane, execution/Edge, native experience, and contract stewardship.